

REMARKS

I. INTRODUCTION

In response to the Final Office Action dated May 22, 2001, this divisional application is being submitted. Applicant traverses the prior rejection. Claims 12-35 remain in the application. Entry of these amendments, and re-consideration of the application, as amended, is requested.

II. ELECTION/RESTRICTION REQUIREMENT

In paragraph (3) of the Office Action, provided that newly submitted claims 12-35 were drawn to a completely different invention that originally presented and therefore, the invention was deemed constructively elected by original presentation for prosecution on the merits.

The rejection stated:

The original presentation of claims 1-11 were directed to an applet receiving commands and using AWT with standard functionality. In stark contrast the newly presented claims 12-35 are directed to a browser application on a computer executing an applet where the applet initiates two different windows outside the browser and uses classes. Further, the window displayed looks like a real application. The first invention was an applet using a tool kit and the second invention is directed to browser application with a single applet generating multiple simultaneous windows with classes and windows that look like a real application.

Applicant traverses such a rejection. The new independent claims are directed towards a browser application and an applet executing therein that displays windows and makes a window look like an executing application (previously submitted new claims were directed towards a looking like a real application).

However, such an invention is not entirely different from the original claims. In the Examiner's own words, the original claims were directed to an applet receiving commands and using AWT with standard functionality. AWT is a class that provides functionality. The new claims specifically provide for the use of an applet and a class that displays a window. Dependent claim 13 provides that the class is the abstract windows toolkit™ (AWT) class. Accordingly, the new claims are directed towards an applet that utilizes a class (AWT) to displays windows. Removing the AWT functionality from the independent claims (as in the original claims) and adding them to a dependent claim (as in the newly submitted claims) where similar functionality is being offered by both sets of claims does not render the claim sets distinct and directed to different inventions.

Accordingly, based on the similarities between the claim sets, the rejection of election by original presentation was improper. Nonetheless, to expedite prosecution, Applicant is filing this divisional application while traversing the election requirement. Further, the detailed response below that was transmitted in response to the first Office Action should be considered when examining the application in view of the previously cited art.

### III. PRIOR ART REJECTIONS UNDER 35 U.S.C. §102(e)

In paragraph (6) of the first Office Action, claims 1-11 were rejected under 35 U.S.C. §102(e) as being anticipated by Craig et al., U.S. Patent No. 6,108,687 (Craig), Dale et al., U.S. Patent No. 6,049,664 (Dale), Faustini, U.S. Patent No. 6,044,218 (Faustini), Van Hoff, U.S. Patent No. 5,802,530 (Van Hoff), and Purnaveja et al., U.S. Patent No. 6,006,241 (Purnaveja) and are rejected under 35 U.S.C. §102(b) as being clearly anticipated by Begole et al., 1997 (Begole) and Lee et al., 1996 (Lee).

The cited references do not teach nor suggest Applicant's independent claims for one or more of the following reasons:

- (1) *The references fail to teach, disclose, or suggest the simultaneous display of multiple windows using a single applet executed by a browser; and*
- (2) *The references fail to teach, disclose, or suggest, simultaneously displaying multiple windows based on a class that comprises elements that make the windows appear like an executing application.*

Applicant has canceled claims 1-11 and added new claims 12-35. Independent claims 12, 20, and 28 are generally directed to an applet (executing in a browser application) that is configured to display multiple windows simultaneously wherein the displayed windows are outside of browser application's window constraints. Further, the applet displays such windows using a class that has elements that cause the window displayed to appear like an executing application.

The cited references provide general references to the use of an applet and various commands for controlling an applet. However, none of the cited references teach utilizing a particular class to display multiple windows that are outside of the browser window's constraints.

Craig merely describes a system disclosed for providing synchronized presentation of slides over a computer network. The display of multiple windows by a single applet as claimed is not taught, disclosed, or suggested.

Dale merely describes a tier-neutral development mechanism for network-based applications. Similar to Craig, Dale fails to teach, disclose, or suggest, the display of multiple windows by a single applet as claimed.

Faustini merely describes a system for a live applet or application development environment includes software that cooperatively promotes and permits immediate socialization of new components with existing components as the new components are instantiated or dropped onto the development desktop. However, similar to Dale and Craig, Faustini fails to teach, disclose, or suggest the use of a particular class that causes a window to be displayed that appears like an executing application.

Van Hoff merely describes a web document based GUI for use on a client computer that is networked with server computers. The Web browser in Van Hoff displays a corresponding GUI web document by executing each of the one or more applets of the corresponding GUI Web document and displaying on the client computer the corresponding interactive image and by displaying on the client computer the one or more links of the corresponding GUI Web document. Thus, multiple applets are used. However, like Craig, Dale and Faustini, Van Hoff lacks any discussion about the use of a single applet to display multiple windows that appear like executing applications.

Purnaveja merely describes the production of synchronization scripts and associated annotated multimedia streams for servers and client computers coupled to each other by a diverse computer network which includes local area networks (LANs) and/or wide area networks (WANs) such as the internet. However, similar to the above discussed references, Purnaveja lacks any discussion about the use of a single applet to display multiple windows that appear like executing applications.

Lee provides support for “multi-user, multi-applet workspaces in CBE”. It can be seen from the title alone that Lee utilizes multiple applets to accomplish any goals. Accordingly, Lee fails to teach, disclose, or suggest the use of a single applet to display multiple windows that appear like executing applications.

Begole provides for the transparent sharing of Java applets. However, similar to all of the above described references, Begole fails to teach, disclose, or suggest the use of a single applet to display multiple windows that appear like executing applications.

In addition to each of the above references failing to teach, disclose, or suggest the invention as claimed, the references in fact teach away from the present invention. For example, whenever one of the cited references provide and utilize multiple applets to display a window, such references teach away from the use of a single applet to display multiple windows.

Moreover, the various elements of Applicant's claimed invention together provide operational advantages over Craig, Dale, Faustini, Van Hoff, Purnaveja, Begole, and Lee. In addition, Applicant's invention solves problems not recognized by Craig, Dale, Faustini, Van Hoff, Purnaveja, Begole, and Lee.

Thus, Applicant submits that independent claims 12, 20, and 28 are allowable over Craig, Dale, Faustini, Van Hoff, Purnaveja, Begole, and Lee. Further, dependent claims 13-19, 21-27, and 29-35 are submitted to be allowable over Craig, Dale, Faustini, Van Hoff, Purnaveja, Begole, and Lee in the same manner, because they are dependent on independent claims 12, 20, and 28, respectively, and thus contain all the limitations of the independent claims. In addition, dependent claims 13-19, 21-27, and 29-35 recite additional novel elements not shown by Craig, Dale, Faustini, Van Hoff, Purnaveja, Begole, and Lee.

IV. CONCLUSION

In view of the above, it is submitted that this application is now in good order for allowance and such allowance is respectively solicited. Should the Examiner believe minor matters still remain that can be resolved in a telephone interview, the Examiner is urged to call Applicant's undersigned attorney.

Respectfully submitted,

Nga T. Dang

By his attorneys,

GATES & COOPER LLP

Howard Hughes Center  
6701 Center Drive West, Suite 1050  
Los Angeles, California 90045  
(310) 641-8797

Date: 8/22/01

By: George H. Gates  
Name: George H. Gates  
Reg. No.: 33,500

## APPENDIX A: SPECIFICATION PARAGRAPHS IN MARKED-UP FORM

Please replace the paragraph beginning at page 1, line 6 with the following:

This invention relates in general to generating Web content, and in particular, to a method for implementing a multiple screen Java™ Applet for use with Web pages.

Please replace the paragraph beginning at page 1, line 10 with the following:

As the popularity and usefulness of the Internet grows, more developers are turning to Java™. Java™ is a popular computer language used to create platform-independent applications. Generally, Java™ applets are loaded into browsers from web pages to create text, graphics, or to perform some other useful function. Within the Java™ programming language, a developer can create applets or standalone applications. Applets are programs executed as part of a Web page and displayed within a Java™-enabled browser. Standalone applications, on the other hand, are general-purpose Java™ applications that don't need a browser to run, but require a Java™ Virtual Machine (JVM) on the computer where the standalone application is to be executed.

Please replace the paragraph beginning at page 1, line 18 with the following:

In the prior art, Java™ applets can only display a single page or window at a time. Generally, this is a restriction imposed by the browser or the browser's JVM. It can be seen then, that there is a need for a method for implementing applets that can display more than one page or window at a time.

Please replace the paragraph beginning at page 3, line 12 with the following:

An object of the present invention is to provide a method for implementing a multiple screen Java™ applet. A further object of the present invention is to provide multiple open applets for simultaneous use on a computer system.

Please replace the paragraph beginning at page 5, line 15 with the following:

Within the Internet environment, the use of Java™ to create web pages and other web-based applications is widespread. Java™ is used because of its platform independence. Theoretically, a Java™ application can be executed by any client computer that has a JVM, regardless of that client computer's hardware or operating system type.

Please replace the paragraph beginning at page 6, line 1 with the following:

A number of high-level features of the Java™ computer language are used to support this platform independence. One such high-level feature is the Abstract Window Toolkit™ (AWT). The AWT is a Graphical User Interface (GUI) that offers a variety of tools for creating buttons, list boxes, etc., drawing two dimensional entities, creating text fonts and colors, scaling entities, etc. Further, the AWT handles events between users and computer systems such as keyboard entries, mouse clicks, etc. AWT provides a foundation for users to make applications and applets that are portable across platforms, which makes the applet or application have similar characteristics, also known as a "look and feel," on whatever platform a particular user is operating.

Please replace the paragraph beginning on page 7, line 13 with the following:

The limitations of the prior art are minimized by the present invention. The present invention implements an applet that intelligently responds to the open and close commands, e.g., the Init, Start, Stop, and Destroy methods implemented in the Java™ computer language, to allow for multiple windows to be open at any given time.

Please replace the paragraph beginning on page 8 line 6 with the following:

Either or both of the Web browser 104 and Web daemon 108 may include a Java™ Virtual Machine (JVM) 110 that executes Java™ applets 112, objects, scripts, etc., associated with various Web content.

Please replace the paragraph beginning on page 9, line 6 with the following:

Within the Java™ language, the AWT offers two classes that can create popup windows that appear outside the constraints of the normal browser 104 area allocated to an applet 112: Window and Frame.

Please replace the paragraph beginning on page 10, line 4 with the following:

The present invention uses the Frame class to allow multiple window applets 112, offered within the Java™ AWT, to make the applet 112 "look and feel" like a Java™ application. Since Frame class objects can be displayed or closed at any time, be minimized or maximized, and displayed at multiple locations on the monitor, the use of the Frame class by the present invention allows multiple window Java™ applets 112 to be open simultaneously, and also allows multiple applets 112 to be open simultaneously.

Please replace the paragraph beginning on page 16, line 15 with the following:

This concludes the description of the preferred embodiment of the invention. The following describes some alternative embodiments for accomplishing the present invention. For example, any type of computer, such as a mainframe, minicomputer, or personal computer, could be used to

implement the present invention. In addition, the present invention is not limited by specific document or programming languages, and could comprise languages other than XML and Java<sup>TM</sup>. For example, the present invention could also be used with HTML, SGML, NetRexx, VisualBasic Script, XML, Perl, C, C++, Cobol, etc